Maryland Computer Science
5th Grade Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The Maryland Computer Science 5th Grade Course introduces students to foundational programming concepts through a block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. Required lessons are labeled with the specific standards they address to support planning and compliance.

Standards Alignment Note: Lessons that list "*Standards Met*" (below) are required to fully meet state computer science standards. Lessons without a standards tag support spiral review, practice, or enrichment.

Programming Environment: Students will write and run programs that are saved in the CodeHop platform. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Prerequisites: There are no prerequisites for this course. It is designed to support all learners, regardless of prior computer science experience.

More Information: Browse the content of this course at https://codehs.com/course/27669/overview

A clickable PDF can be found at https://codehs.com/MD-CSRoadmaps

Course Breakdown

Optional Review

In this foundational unit, students get reacquainted with CodeHop and begin to build familiarity with the coding interface, coordinate systems, and creative tools.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Understand computer science terms and programming basics. ● Explore the coordinate plane and drawing tools.
Lessons	**Welcome to CodeHop!** ● Practice logging in and exploring the Playground before starting a full lesson. **Introduction to Computer Science and Scratch** ● Review basic computer science vocabulary and create a simple program. **The Coordinate Plane** ● Use the coordinate plane to design an open-ended animation. **Drawing Tools** ● Create customized sprites and backdrops using the drawing tools.

Unit 1: Getting Started (2 lessons)

Students are introduced to basic computing systems and computational thinking practices that lay the foundation for later programming and problem-solving activities.

Objectives / Topics Covered	● Identify parts of a computing system. ● Distinguish between hardware and software. ● Recognize and describe simple hardware and software problems. ● Apply computational thinking to design and plan solutions.
Lessons	**Practicing with Computing Systems** ● Identify parts of a computing system and basic troubleshooting steps. ● *Standards Met: 5.CS.D.01, 5.CS.HS.01, 5.CS.T.01* **Computational Thinking: Design a Neighborhood** ● Apply computational thinking to plan and design a neighborhood layout. ● *Standard Met: 5.AP.M.01*

Unit 2: Sequences, Events & Loops (5 lessons)

Students build foundational programming skills by creating and refining algorithms, using events and loops to control programs, debugging errors, and exploring creative tools within the programming environment.

Objectives / Topics Covered	● Use sequences, events, and loops to control program behavior. ● Create and refine algorithms to solve problems efficiently. ● Debug programs by decomposing steps and identifying errors. ● Use drawing and pen tools to create visual programs. ● Apply looping patterns to create repeated designs and animations.
Lessons	**Drawing Tools: Sea Creatures** ● Use image editing tools to create and program deep-sea creatures. **Events: Traveling with Scout** ● Use event blocks to trigger character actions in a travel-themed program. **Compare and Refine Algorithms** ● Evaluate and improve multiple algorithms for a task. ● *Standard Met: 5.AP.A.01* **Debugging: Make a Pizza**

	● Break down and fix a pizza-making program to ensure it runs correctly. ● *Standard Met: 5.AP.PD.03* **Creating Turtle Graphics** ● Use loops and the pen tool to draw repeated, artistic patterns.

Unit 3: Conditionals & Operators (5 lessons)

Students develop decision-making skills in programs by using conditionals, comparison operators, and variables to control behavior and create interactive game mechanics.

Objectives / Topics Covered	● Use conditionals to make decisions in a program. ● Explain and apply if/then/else logic. ● Use comparison operators and variables to control outcomes. ● Plan and decompose steps needed to build an interactive program.
Lessons	**Conditionals: Mazes** ● Create a maze program that uses conditionals to navigate paths. **Complex Conditionals: Chase the Star** ● Build a game using “if/then/else” blocks for branching decisions. **Game Mechanics with Comparison Operators** ● Use comparison operators and variables to create ending game mechanics. **Plan a Quest (2 day lesson)** ● Break down and plan the steps needed to build a quest-style program.

Unit 4: Variables & Lists (4 lessons)

Students expand their data-handling skills by using variables, operators, and lists to store, update, and manipulate information within interactive programs.

Objectives / Topics Covered	● Use variables to store and change values in a program. ● Apply operators to modify variable values. ● Use lists to organize and manage collections of data. ● Combine variables, operators, and lists to model real-world scenarios.
Lessons	**Variables in Dance** ● Use variables to control pitch and dance speeds in a program. ● *Standard Met: 5.AP.V.01* **Operators: Coin Flip** ● Create a coin flipping program using variables and operators. ● *Standard Met: 5.AP.C.01* **Lists: Shopping with Scout (2 day lesson)** ● Create a shopping simulator using variables, lists, and operators.

Unit 5: Culmination Projects (8 lessons)

Students apply advanced programming concepts and design thinking to create, analyze, and refine interactive projects. Through game design, app development, and data inquiry, students demonstrate creativity, problem-solving, and an understanding of real-world computing applications.

Objectives / Topics Covered	● Give proper attribution when creating or remixing programs. ● Analyze and improve programs by modifying and extending code. ● Use clones to create multiple interacting objects. ● Apply classes and objects to structure game programs. ● Use design thinking to plan and design an app that meets user needs.

	● Follow an inquiry process to analyze data and present results visually.
Lessons	**Giving Credit Through Attributions** ● Give appropriate attribution when creating or remixing programs online. ● *Standard Met: 5.AP.PD.02* **Careers in CS: Mobile Apps** ● Analyze and improve a game by modifying code and adding new functionality. ● *Standard Met: 5.AP.PD.04* **Clones: Throwing Acorns Game** ● Create a game where acorns are cloned and thrown toward targets. **Classes and Objects in Games (2 day lesson)** ● Create an interactive game using classes, objects, and randomizers to vary object behavior. **Design an App (3 day lesson)** ● Use design thinking to plan an app that solves a real-world problem. ● *Standards Met: 5.IC.SI.02, 5.IC.C.02* **Inquiry Project: Line Graph (2 day lesson)** ● Investigate a question and modify a program to present the results as a line graph.

Unit 8: Digital Literacy (10 lessons)

All lessons in this unit are required for full standards alignment. Students develop advanced digital literacy skills by learning how to behave responsibly online, protect personal information, collaborate digitally, analyze data, understand cybersecurity and networks, and examine the societal impacts of computing and innovation.

Objectives / Topics Covered	● Understand digital footprints and practice appropriate digital etiquette. ● Identify personally identifiable information and explain why it must be protected. ● Practice safe and responsible online behavior, including reporting concerns. ● Collaborate with others using digital tools to improve programs. ● Explore cybersecurity concepts and privacy protection. ● Search for information online and provide proper attribution. ● Analyze data to draw conclusions and make predictions. ● Explain how digital data is stored and managed. ● Describe how networks and protocols enable data transfer. ● Examine how technology and culture influence each other over time.
Lessons	**Digital Footprint & Digital Etiquette** ● Identify personally identifiable information, understand digital footprints, and practice respectful and responsible online behavior. ● *Standard Met: 5.NI.C.01* **Online Safety** ● Develop and practice a code of conduct for online participation, including identifying and reporting inappropriate behavior. ● *Standard Met: 5.IC.SI.01* **Collaborating Globally (2 day lesson)** ● Collaborate digitally with others to improve a program. ● *Standard Met: 5.IC.SI.02* **Cybersecurity & Privacy** ● Create a program that demonstrates how to solve a real-world cybersecurity issue. ● *Standard Met: 5.NI.C.02* **Use and Search the Right Way** ● Search online effectively and provide proper attribution to sources. ● *Standard Met: 5.IC.SLE.01* **Data Analysts** ● Evaluate and analyze data for reliability to draw conclusions and make predictions. ● *Standards Met: 5.DA.CVT.01, 5.DA.IM.01* **File Management and Data Exploration**

	• Explain how digital data varies in size and where it can be stored. • *Standard Met: 5.DA.S.01* **Networks and Protocols** • Explain how network protocols enable data transfer and compare WiFi, wired, and cellular networks. • *Standard Met: 5.NI.NCO.01* **Impacts of Computing: Innovation** • Explore how computing innovations have changed the way we live and work. • *Standards Met: 5.IC.SI.02, 5.IC.C.01*

Maryland Computer Science 5th Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
Warm-Up Activities	This warm-up activity slide deck provides 5-10 minute problems aligned with computer science skills to engage students at the start of class, allowing teachers to preview or review concepts with answer keys and discussion tips included in the Speaker Notes.
Program Self-Assessment (Spanish)	This is a student self-assessment tool designed to help K-6 learners reflect on their programming projects, evaluate their skills in algorithms, debugging, collaboration, and reflection, and set goals for improvement.
Peer Review Resources (Spanish)	This provides structured worksheets to facilitate student feedback during collaborative coding projects. It encourages reflection by guiding students to highlight successes, ask questions, and offer constructive feedback on their partner's work.
Lesson Reflection & Computational Thinking (Spanish)	This guides students in engaging with computational thinking concepts, preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
Design-Your-Own-Lesson Templates	Empower your students to explore and express their knowledge creatively with our versatile graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
All of these resources and more are found on the **CodeHop Resources Page**.	